ISEP
Machine Learning
November 2nd 2020

Unsupervised learning : Clustering

Graded Lab

1 Part I : Exercises

Exercise 1. Suppose that we have four observations, for which we compute a dissimilarity matrix, given by :

$$\begin{bmatrix} & 0.3 & 0.4 & 0.7 \\ 0.3 & & 0.5 & 0.8 \\ 0.4 & 0.5 & & 0.45 \\ 0.7 & 0.8 & 0.45 & \end{bmatrix}$$

For instance, the dissimilarity between the first and second observations is 0.3, and the dissimilarity between the second and fourth observations is 0.8.

(a) On the basis of this dissimilarity matrix, sketch the dendrogram that results from hierarchically clustering these four observations using complete linkage. Be sure to indicate on the plot the height at which each fusion occurs, as well as the observations corresponding to each leaf in the dendrogram.

(b) Repeat (a), this time using single linkage clustering.

(c) Cut the dendrogram obtained in (a) such that two clusters result. Which observations are in each cluster ?

(d) Cut the dendrogram obtained in (b) such that two clusters result. Which observations are in each cluster ? Compare the results with those obtained in the previous question.

Exercise 2. In this problem, you will perform K-means clustering manually, with $K = 2$, on a small example with $n = 6$ observations and $p = 2$ features. The observations are as follows.

Obs	X_1	X_2
1	1	4
2	1	3
3	0	4
4	5	2
5	6	2
6	4	0

(a) Plot the observations.

(b) Randomly assign a cluster label to each observation. You can use the `choice()` command from the `random` module in `Python` or `choices()` for Python versions 3.6 and up. Report the cluster labels for each observation.

(c) Compute the centroid for each cluster.

(d) Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation. For that, plot the observations and the centroids.

(e) Repeat (c) and (d) until the answers obtained stop changing.

(f) Finally, In your plot, color the observations according to the cluster labels obtained.

2 Part II : Practical applications

2.1 K-means clustering

In Python you can use the function KMeans() from the module sklearn.cluster to perform K-means clustering. To begin you will perform K-means with simulated data. Follow the steps :

(a) The simulated data will consist in 50 observations described by two normal-distributed variables. In order to define classes in the data the first 25 observations have a mean shift relative to the next 25 observations.

```
import numpy as np
X = np.random.randn(50,2)
X[0:25, 0] = X[0:25, 0] + 3
X[0:25, 1] = X[0:25, 1] - 4
```

You can plot the observations and notice that there are two well separated clusters :

```
import matplotlib.pyplot as plt
plt.plot(X[:,0], X[:,1],"o")
plt.xlabel("X1")
plt.ylabel("X2")
```

(b) Perform K-means clustering with $K = 2$. Use the following command :

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = 2, random_state = 100).fit(X)
```

Then, the cluster assignments can be obtained by running print(kmeans.labels_). The final clusters centroids can be known using the command print(kmeans.cluster_centers_).

In addition, you can plot the observations using a different color per cluster by running the following code :

```
for i in range(0,X.shape[0]):
    if kmeans.labels_[i] == 0:
        plt.plot(X[i:,0], X[i:,1],"o",color= "red")
    else:
        plt.plot(X[i:,0], X[i:,1],"o",color= "blue")
centroid1=kmeans.cluster_centers_[0]
centroid2=kmeans.cluster_centers_[1]
plt.plot(centroid1[0],centroid1[1],"*",color= "black",markersize = 15)
plt.plot(centroid2[0],centroid2[1],"*",color= "black",markersize = 15)
plt.xlabel("X1")
plt.ylabel("X2")
```

(c) [graded question] Now, you are going to perform $K-$means with real data. The file *HCVdata.csv* contains laboratory values of blood donors and Hepatitis C patients and demographic values like age. There are 12 variables, features 4-12 concern laboratory data (ALB, ALT, AST, BIL, CHE, CHOL, CREA, GGT and PROT). Import the data set and get familiar with the data. Answer the following questions : how many observations are there ? How many variables are there ? What type of variables (numeric, categorical) ? For each categorical variable (if any) give the number of levels and categories. You will denote the resulting dataframe object dfHVC.

(d) [graded question] Since all the clustering approaches used in this lab are based upon distances or dissimilarity measures, it is strongly recommended to scale the variables previously. The following code allows to scale all the variables in a dataframe called dfHVC :

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
dfHVC_scaled = scaler.fit_transform(dfHVC)
```

Create a dataframe containing all the numeric variables after being scaled, you will name the resulting output `dfHVC_scaled`.

(e) [graded question] The purpose of this study is to determine if it is possible to get clusters of patients by considering blood laboratory analysis and the age. Perform K-means clustering with $K = 4$ considering only *Age* and all the variables concerning laboratory data. This time you will set the parameter n_init of the function `KMeans()` to 50. This parameter allows to run the Kmeans() function with multiple initial cluster assignments (default : 10) and then report only the best results. In addition , set the `random_state` parameter to 1000.

```
dfHVC_Kmeans4=KMeans(n_clusters = 4, n_init = 50, random_state =
    1000).fit(dfHVC\_scaled)
```

Although, we are performing unsupervised learning, we can use the variable category to interpret the obtained clusters. To this end, you can calculate a matching matrix, which is a special kind of confusion matrix, with two dimensions (”real groups” versus ”clusters”). The variable *Category* contains the real groups of patients (`0=Blood Donor`, `1=Hepatitis`, `2=Fibrosis` and `3=Cirrhosis`). Use the following code to calculate the matching matrix and interpret the results by trying to match each real group to a given cluster detected by the algorithm.

```
pd.crosstab(index = dfHVC_labels,
            columns = dfHVC_Kmeans4_labels,
            rownames = [’Real␣groups’],
            colnames = [’K–means␣clusters’])
```

where `dfHVC_labels` are the labels defined by the variable *Category* and `dfHVC_Kmeans4_labels` are the cluster labels returned by the `KMeans()` function and `pd` is the `pandas` library.

(f) [graded question] According to the results obtained in the previous question, do you think it is suitable to run $K-$means with another value of K ? If so, perform once more K-means with this value and interpret the results.

2.2 Hierarchical clustering

In Python, the `linkage()` function from `SciPy` implements hierarchical clustering . For instance, in order to perform hierarchical clustering using complete linkage for the simulated data generated in the previous section, you can execute the following code :

```
from scipy.cluster.hierarchy import linkage
hc_complete = linkage(X, ”complete”)
```

Then, in order to plot the associated dendrogram using the previous results, you can use the code :

```
from scipy.cluster.hierarchy import dendrogram
import matplotlib.pyplot as plt
plt.figure(figsize=(25, 10))
dendrogram(hc_complete)
plt.show()
```

We remark the 2 clusters are clearly identified, which is normal because we performed hierarchical clustering with simulated data.
To determine the cluster labels for each observation associated with a given cut of the dendrogram, you can use the `cut_tree()` function from the `scipy.cluster.hierarchy` module . For example, to obtain two clusters you can execute the code :

```
from scipy.cluster.hierarchy import cut_tree
print(cut_tree(hc_complete, n_clusters = 2).T)
```

(a) [graded question] Perform hierarchical clustering using the `HCVdata` using complete, average and single linkage. To begin you will cut the dendrogram at 4 clusters. Similarly to the previous section, calculate the matching matrix and comment the results.

(b) [graded question] Plot the dendrogram. Do you think is it suitable to cut the dendrogram at a level different from 4 clusters ? If so, why ? Calculate the matching matrix for this chosen number of clusters and interpret the results.

References

— James, Gareth ; Witten, Daniela ; Hastie, Trevor and Tibshirani, Robert (2013). ”An Introduction to Statistical Learning with Applications in R”. New York : ”Springer texts in statistics”. Site web : `http://www-bcf.usc.edu/~gareth/ISL/`

— J Crouser : ”SDS 293 - Machine Learning labs”. `http://www.science.smith.edu/~jcrouser/SDS293/`. Visited on October 18th.

— Lichtinghagen R et al. J Hepatol 2013 ; 59 : 236-42.

— Hoffmann G et al. Using machine learning techniques to generate laboratory diagnostic pathways - a case study. J Lab Precis Med 2018 ; 3 : 58-67